CPE301 – SPRING 2022

Design Assignment 4

Student Name: Ernesto Ibarra-Ayala
Student #: 2001211571
Student Email: ibarre3@unlv.nevada.edu
Primary Github address: https://github.com/ErnestoIbarra333/ErnestoIbarra

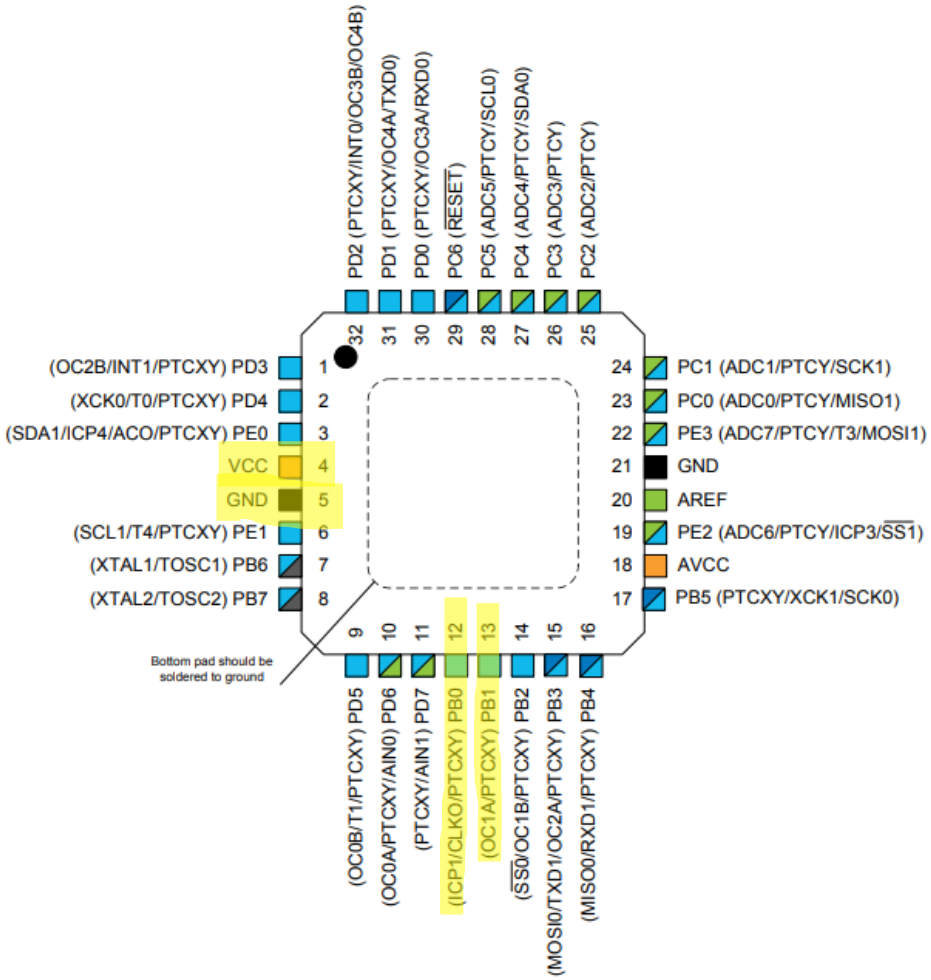
1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

Atmel Studio 7.0
- Assembler
- Simulator
- Debugger

Atmega328PB-Xmini PC

Multi-Function Shield
- Switches
- LEDs

Logic Analyzer

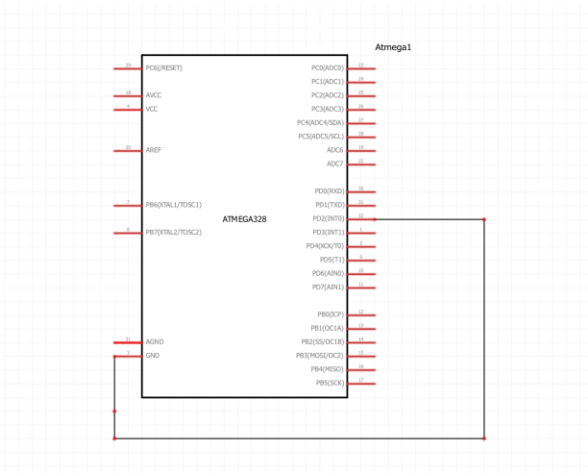


2. DEVELOPED MODIFIED CODE OF TASK 1/2/3

```
#define F_CPU 16000000UL
#include <stdio.h>
#include <stdlib.h>
#include <avr/io.h>
#include <string.h>
#include <util/delay.h>
#include <avr/interrupt.h>

#define BAUDRATE 9600
#define BAUD_PRESCALLER (((F_CPU / (BAUDRATE * 16UL))) - 1)
#define  T_pin PB1 // pin for trigger
int TimerOF = 0; // overflow counter

int main(void)
{
    char distance_string[10];
    long count2;
    double distance;

    DDRB = 0x02; // set PB1 and PB2 as outputs
    USART_init(); // initialize the USART
       TCCR1A = 0;  // start in normal mode
    TIMSK1 = (1 << TOIE1); // enable timer 1 overflow interrupt
    sei(); // enable interrupts

    while(1)
    {

        PORTB |= (1 << T_pin);
        _delay_us(10); // add a quick trigger pulse for trigger pin
        PORTB &= (~(1 << T_pin));

        TCNT1 = 0; // start timer at 0
        TCCR1B = 0x41; // capture rising edge and with no pre scalar
        TIFR1 = 1<<ICF1; // clear ICP flag
        TIFR1 = 1<<TOV1; // clear overflow flag

        while ((TIFR1 & (1 << ICF1)) == 0); //We stay here until rising edge
        TCNT1 = 0; // start timer at 0
        TCCR1B = 0x01; // capture falling edge instead now, no pre scalar
        TIFR1 = 1<<ICF1; // clear ICP flag
        TIFR1 = 1<<TOV1; // clear overflow flag
        TimerOF = 0; // clear our overflow timer

        while ((TIFR1 & (1 << ICF1)) == 0); // we stay until falling edge
        count2 = ICR1 + (65535 * TimerOF); // receive value from capture register
        /* 8MHz Timer freq, sound speed = 343 m/s, calculation mentioned in doc. */
        distance = (double) count2 / (933);

        dtostrf(distance, 2, 2, distance_string); // here we will convert our distance
into a string to display it later
        strcat(distance_string, " cm   ");
        USART_putstring("Dist = ");
        USART_putstring(distance_string); // here we use our function put string to
display it in our terminal
        USART_putstring("\n");
```

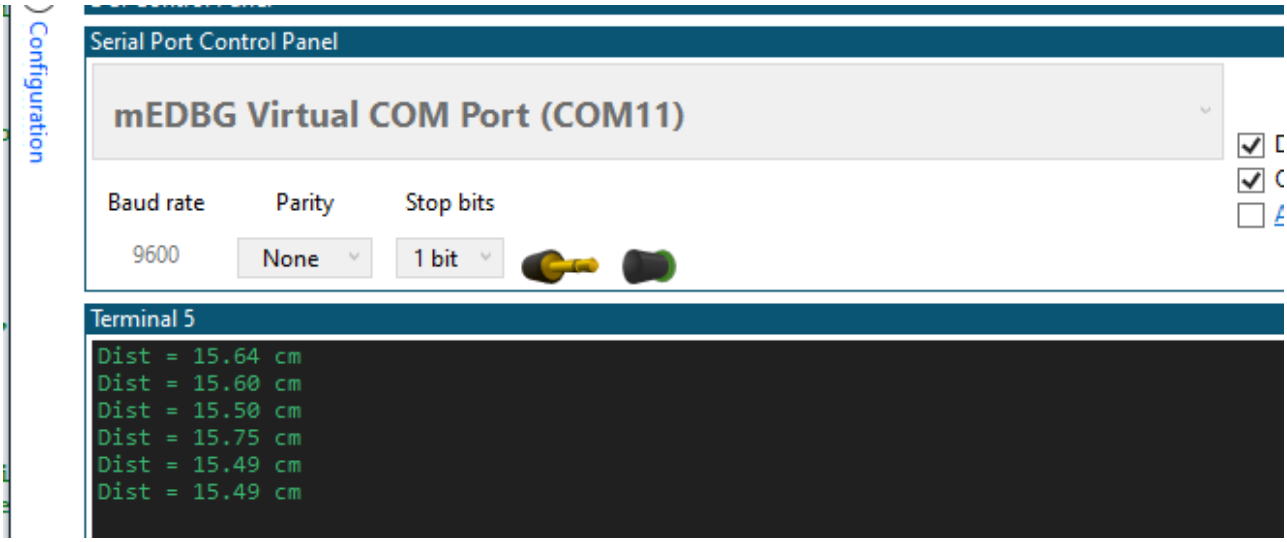
```
        _delay_ms(1000); // use a delay of 1 second as instructed.
    }
}

ISR(TIMER1_OVF_vect)
{
       TimerOF++; // here we increment our counter
}

void USART_init(void)
{

       UBRR0H = (uint8_t)(BAUD_PRESCALLER>>8);
       UBRR0L = (uint8_t)(BAUD_PRESCALLER);
       UCSR0B = (1<<RXEN0)|(1<<TXEN0);
       UCSR0C = (3<<UCSZ00);
}

void USART_send( unsigned char data)
{

       while(!(UCSR0A & (1<<UDRE0)));
       UDR0 = data;

}

void USART_putstring(char* StringPtr)
{

       while(*StringPtr != 0x00)
       {
              USART_send(*StringPtr);
              StringPtr++;
       }

}
```

3. SCHEMATICS



4. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

Here we can see the main screenshot for Task 1 of our assignment. We needed to run a program to measure the distance using the HC-SR04 ultrasonic module. Here is a screenshot of the output of the ultrasonic module on the atmel terminal.

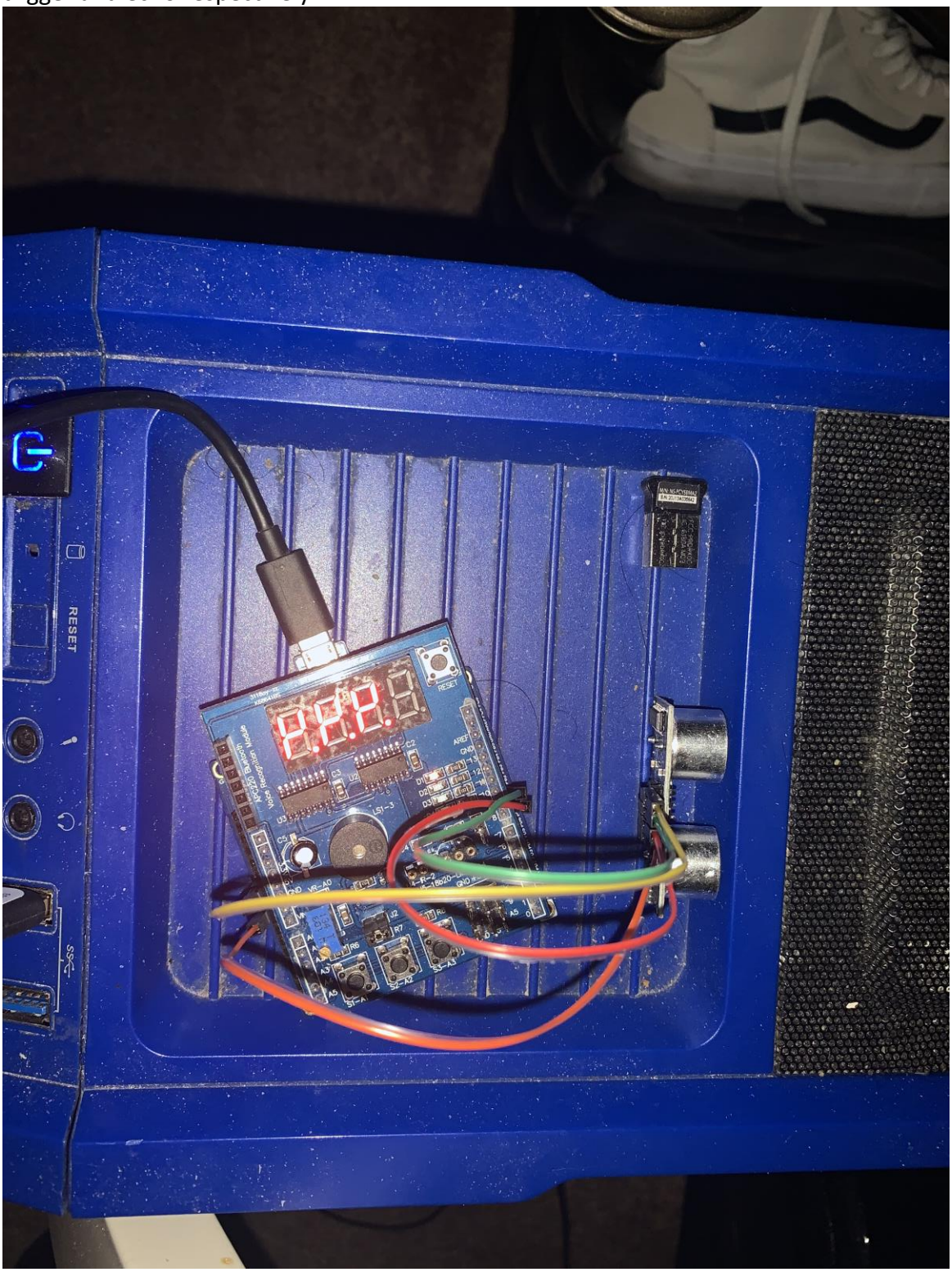


For task two and three it says to use the Timer Capture Function to determine the ECHO capture pulse. We used that formula to get an accurate distance measured, here is screenshot of that formula on my code. Also for task three it asked us to display this information every 1 second which is also shown in the screenshot below as well as in the screenshot above.

```
while ((TIFR1 & (1 << ICF1)) == 0); // we stay until falling edge
count2 = ICR1 + (65535 * TimerOF); // receive value from capture register
distance = (double) count2 / (933);

dtostrf(distance, 2, 2, distance_string); // here we will convert our distance into a string to display it lat
strcat(distance_string, " cm   ");
USART_putstring("Dist = ");
USART_putstring(distance_string); // here we use our function put string to display it in our terminal
USART_putstring("\n");
_delay_ms(1000); // use a delay of 1 second as instructed.
```

5. SCREENSHOT OF EACH DEMO (BOARD SETUP)

Here we can see how we connected our Ultrasonic Module to our atmega328pb. I am using VCC and ground which are the orange and yellow wires. I'm also using PB0 and PB1 for the trigger and echo respectively.



6. VIDEO LINKS OF EACH DEMO

Below is the video for task 1/2/3
https://youtu.be/RqSpLG3XfbI

7. GITHUB LINK OF THIS DA

https://github.com/ErnestoIbarra333/ErnestoIbarra/tree/main/Design%20Assignments

Student Academic Misconduct Policy
http://studentconduct.unlv.edu/misconduct/policy.html

"*This assignment submission is my own, original work*".
Ernesto Ibarra